Analyzing the Indexing Effect on Stock Returns – Evidence from the German Stock Market

Jonas Furrer

11/05/2023

Generalized Synthetic Control Method Analysis

Loading packages.

```
library(PerformanceAnalytics)
library(tidyverse)
library(quantmod)
library(ggplot2)
library(purrr)
library(lubridate)
library(rlist)
library(grid)
library(gridExtra)
library(plm)
library(tsibble)
library(SciViews)
library(reshape2)
library(gsynth)
library(panelView)
library(lme4)
library(useful)
library(cowplot)
library(ggpubr)
library(data.table)
library(readr)
library(readxl)
library(grDevices)
library(arsenal)
library(reporter)
```

Data Preparation

Basic preparation

Load data with the API from Yahoo! Finance (https://finance.yahoo.com/).

```r
# Defining treatment group and time period (take larger time period to get an overview)
t_seq= c("AIR.DE", "ZAL.DE", "SHL.DE", "SY1.DE", "SRT3.DE", "BNR.DE", "QIA.DE", "PUM.DE", "HFG.DE")
start <- "2019-01-01"
end <- "2022-12-31"

# Load all variables of the treatment group
Stocks_prices_dax_t = lapply(t_seq, function(sym) {
  getSymbols(sym, from=start, to=end, auto.assign=FALSE)
})

# Defining control group
c_seq <- c( "COP.DE", "AIXA.DE", "RAA.DE", "TKA.DE", "VAR1.DE", "COK.DE", "TMV.DE","HNR1.DE","AOX.DE","
            "HLE.DE","EVD.DE","WCH.DE","AT1.DE","AFX.DE","BOSS.DE","NDA.DE","BC8.DE","CBK.DE","DUE.DE",
            "EVK.DE","EVT.DE","FRA.DE","FNTN.DE","FPE3.DE","GXI.DE","GYC.DE","SDF.DE","KGX.DE","LXS.DE"
            "LEG.DE","NEM.DE","PSM.DE","RHM.DE","G24.DE","SOW.DE","SAX.DE","TEG.DE","O2D.DE","UN01.DE",
            "UTDI.DE","G1A.DE")

# Load all variables of the control group
Stocks_prices_dax_c <- lapply(c_seq, function(sym) {
  getSymbols(sym, from=start, to=end, auto.assign=FALSE)
})
```

Create a dummy variable for each element of the list for the treatment and control group.

```r
for (i in 1 : length(Stocks_prices_dax_c)) {
  Stocks_prices_dax_c[[i]]$Treatment <- 0
}

for (i in 1 : length(Stocks_prices_dax_t)) {
  Stocks_prices_dax_t[[i]]$Treatment <- 1
}
```

Rename the elements of both lists according to the defined treatment and control group and combine the two lists.

```r
names(Stocks_prices_dax_c) <- c_seq
names(Stocks_prices_dax_t) <- t_seq

Stock_data <- c(Stocks_prices_dax_c, Stocks_prices_dax_t)
```

Calculate returns based on the closing prices for each list element and rename the variables of each list element.

```r
for (i in 1:length(Stock_data)){
  x <- lapply(Stock_data[[i]][,4], CalculateReturns)
  Stock_data[[i]] <- cbind(as.data.frame(Stock_data[[i]]), as.data.frame(x))
}

colnames <- c("open", "high", "low", "close", "volume", "adjusted", "treatment", "return")
Stock_data <- lapply(Stock_data, setNames, colnames) #change colnames for each list element
```

Import DAX price index returns and create a column with dates.

```r
return_DAX <- lapply("^GDAXIP", function(sym) {
  dailyReturn(na.omit(getSymbols(sym, from=start, to=end, auto.assign=FALSE)))
})

return_DAX <- as.data.frame(return_DAX)
return_DAX <- cbind(date = rownames(return_DAX), return_DAX)
rownames(return_DAX) <- 1:nrow(return_DAX)
return_DAX$date <- as.Date(return_DAX$date)
```

Prepare stock data and the data frame of DAX return for merging.

```r
for (i in 1:length(Stock_data)){
  Stock_data[[i]] <- cbind(date = rownames(Stock_data[[i]]), Stock_data[[i]])
  rownames(Stock_data[[i]]) <- 1:nrow(Stock_data[[i]])
}

return_DAX <- return_DAX %>%
  mutate(date = as.character(date)) %>%
  rename(dax_return = daily.returns)
```

Merge return_DAX to every list element of Stock_data and generate a long format data frame.

```r
for (i in 1:length(Stock_data)) {
  Stock_data[[i]] <- merge(Stock_data[[i]], return_DAX, by = "date", all.x = T)
}

Stock_data <- map_df(Stock_data, ~as.data.frame(.x), .id="ticker")
```

Market-Model and CAR

Calculate the abnormal return using the market model ($r_{it} = \alpha_i + \beta_i * dax\_return_t$) for the CAPM period from 01.07.2020 to 16.08.2021.

```r
# Filter for the relevant time period
Stock_data_DID_CAPM_period <- Stock_data %>%
  filter(date >= as.Date("2020-07-01") & date <= as.Date("2021-08-16")) %>%
  na.omit()

Stock_data_DID_CAPM_period$date <- as.Date(Stock_data_DID_CAPM_period$date)

# Run a regression for each firm and store relevant results in a data frame
reg <- lmList(data=Stock_data_DID_CAPM_period, return ~ dax_return | ticker)
res <- as.data.frame(summary(reg)$coefficients)
res <- res[,c(1,5)]
colnames(res) <- c("alpha","beta")
res$ticker <- rownames(res)

# Merge the results to the initial data set
Stock_data <- left_join(Stock_data, res, by="ticker")

# Calculate abnormal return
Stock_data <- Stock_data %>%
  mutate(AR = return-alpha-beta*dax_return)
```

Calculate the cumulative abnormal return from 01.01.2021 to 08.10.2021.

```r
Stock_data <- Stock_data %>%
  filter(date >= (as.Date("2021-01-01")) & date <= (as.Date("2021-10-08")))

Stock_data <- Stock_data %>%
  group_by(ticker) %>%
  mutate(CAR = cumsum(AR))
```

Create an interaction variable between treatment and period before/after the time of interest and filter data.

```r
Stock_data <- Stock_data %>%
  mutate(period = (ifelse (date >= as.Date("2021-08-16"),1,0))) %>%
  mutate(period_interaction = treatment * period)

Stock_data$date <- as.Date(Stock_data$date)

Stock_data_gsynth_p1_car <- Stock_data %>%
  select(CAR, period_interaction, date, ticker, AR) %>%
  na.omit() %>%
  mutate(CAR_p = CAR*100)
```

Generalized Synthetic Control

Generate a list ("out") with all relevant information and outputs.

For a detailed description of the output variables, see https://cran.r-project.org/web/packages/gsynth/gsynth.pdf.

```r
system.time(
  out <- gsynth(CAR_p ~ period_interaction,
                data = Stock_data_gsynth_p1_car,
                index = c("ticker","date"),
                force = "two-way",
                CV = T,
                r = c(0,22),
                se = TRUE,
                inference = "parametric",
                nboots = 2000,
                parallel = F)
)
```

```
## Cross-validating ...
##  r = 0; sigma2 = 54.26080; IC = 4.25329; PC = 52.70000; MSPE = 30.60947
##  r = 1; sigma2 = 34.96566; IC = 4.07115; PC = 38.07666; MSPE = 24.43523
##  r = 2; sigma2 = 25.11049; IC = 3.99519; PC = 30.30717; MSPE = 19.82450
##  r = 3; sigma2 = 18.59751; IC = 3.94788; PC = 24.64493; MSPE = 13.98293
##  r = 4; sigma2 = 13.05671; IC = 3.84491; PC = 18.84914; MSPE = 11.68167
##  r = 5; sigma2 = 10.35635; IC = 3.86179; PC = 16.18016; MSPE = 9.11124
##  r = 6; sigma2 = 8.57595; IC = 3.91956; PC = 14.41864; MSPE = 8.11204
##  r = 7; sigma2 = 6.93992; IC = 3.95210; PC = 12.49516; MSPE = 6.12692
##  r = 8; sigma2 = 5.68882; IC = 3.99536; PC = 10.92200; MSPE = 5.79361
##  r = 9; sigma2 = 5.06053; IC = 4.11818; PC = 10.32134; MSPE = 5.63651
##  r = 10; sigma2 = 4.39498; IC = 4.21485; PC = 9.49091; MSPE = 5.29980
##  r = 11; sigma2 = 3.86078; IC = 4.32076; PC = 8.80120; MSPE = 5.18711
##  r = 12; sigma2 = 3.42153; IC = 4.43329; PC = 8.21182; MSPE = 4.75785
##  r = 13; sigma2 = 3.01906; IC = 4.53929; PC = 7.61010; MSPE = 4.31505
##  r = 14; sigma2 = 2.77271; IC = 4.68312; PC = 7.32428; MSPE = 4.30770
##  r = 15; sigma2 = 2.54503; IC = 4.82422; PC = 7.03110; MSPE = 4.23487
##  r = 16; sigma2 = 2.33908; IC = 4.96443; PC = 6.74601; MSPE = 4.22319
##  r = 17; sigma2 = 2.15245; IC = 5.10369; PC = 6.46951; MSPE = 4.16659
##  r = 18; sigma2 = 1.99836; IC = 5.24965; PC = 6.24988; MSPE = 3.95703
##  r = 19; sigma2 = 1.85684; IC = 5.39425; PC = 6.03397; MSPE = 3.90911
##  r = 20; sigma2 = 1.72683; IC = 5.53753; PC = 5.82273; MSPE = 3.75820
##  r = 21; sigma2 = 1.59337; IC = 5.67079; PC = 5.56802; MSPE = 3.58346
##  r = 22; sigma2 = 1.47098; IC = 5.80238; PC = 5.32097; MSPE = 3.45022
##
##  r* = 22
##
## Simulating errors .......................Bootstrapping ...
## ....................

##    user  system elapsed
##   11.75    0.87   12.66
```

Creating Plots

Basic data preparation

For figure 2,3 and 5, Y.bar = average values of Y.tr, Y.ct and Y.co, time = date and att = average treatment effect on the treated over time is needed and extracted from “out”.

```
out.dat <- data.frame(ytr=out$Y.bar, time=out$time, att=out$att)
out.dat <- out.dat %>%
  rename(Treatment = ytr.Y.tr.bar,
         Counterfactual = ytr.Y.ct.bar,
         Control = ytr.Y.co.bar)

day <- seq(-172,24,1)
out.dat <- cbind(out.dat,as.data.frame(day))
out.dat <- out.dat %>%
  rename(`Treatment Average` = Treatment,
         `Counterfactual Average` = Counterfactual,
         `Control Average` = Control)
```

Figure 2: Average CAR of the treatment and control group

Data Preparation.

```
figure_DID_d <- out.dat %>%
  pivot_longer(
    cols = c("Treatment Average", "Counterfactual Average", "Control Average", "att"),
    names_to = "category") %>%
  filter(category %in% c("Treatment Average", "Control Average"))
```

```
figure_DID_d$category <- factor(figure_DID_d$category, levels = c("Control Average", "Treatment Average
```

Create figure.

```
ggplot(data=figure_DID_d)+
  geom_line(aes(x=time, y=value, color=category, group=category), lwd=1, key_glyph = "smooth")+
  scale_x_date(date_breaks = "2 weeks" ,date_labels = "%D")+
  geom_vline(aes(xintercept = as.Date(c("2021-09-06")), linetype="Announcement Day"),lwd=1, color = "bla
  geom_vline(aes(xintercept = as.Date(c("2021-09-20")), linetype="Effective Day"),lwd=1,  color = "blac
  ggtitle("")+
  xlab("")+
  ylab("CAR %")+
  labs(color=NULL)+
  scale_color_manual(name = "", values = c(`Control Average` = "grey84", `Treatment Average` ="#3C93C2"
  scale_linetype_manual(name = "", values = c(`Announcement Day` = "solid",`Effective Day` = "dotdash")
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom"
  )+
  guides(color = guide_legend(byrow = T, order = 1, nrow = 2),
         linetype= guide_legend(byrow = T, order = 2, nrow = 2))
```

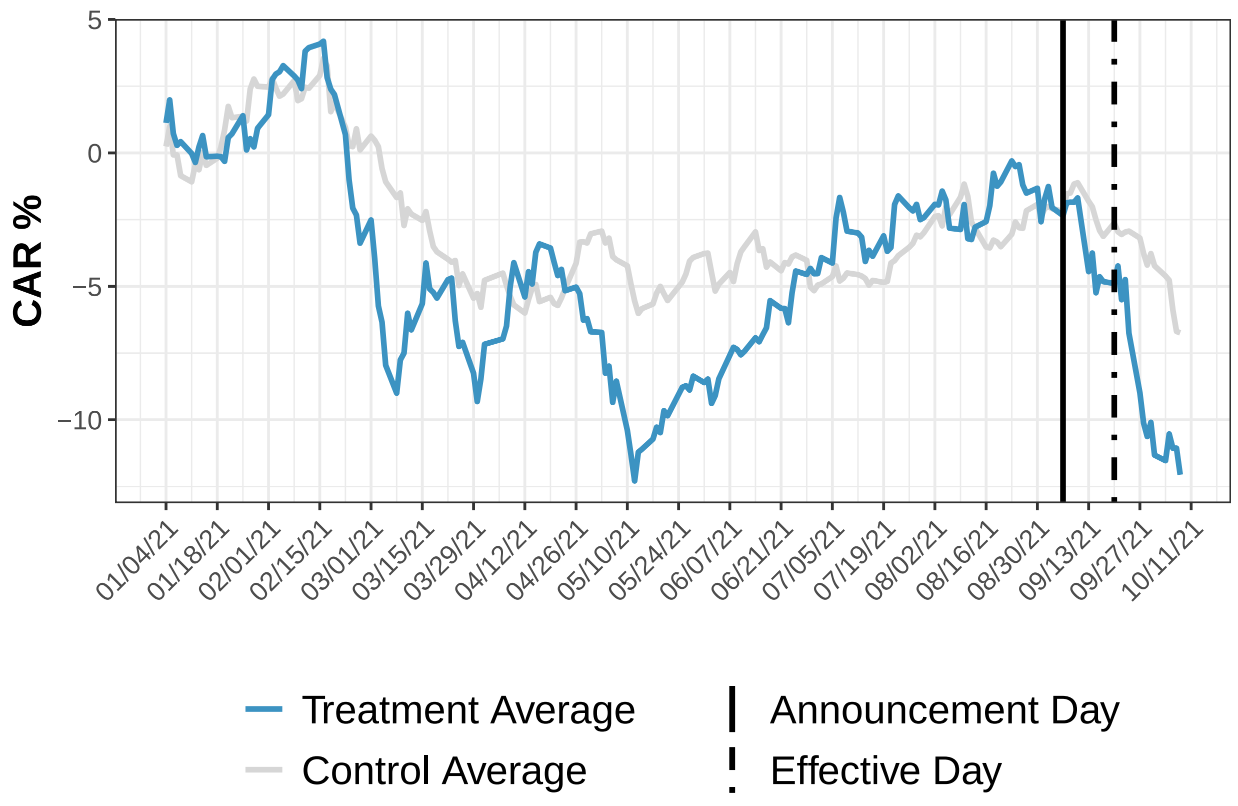


Figure 3: Visual diagnostic test

Data preparation.

```
figure_limit <- out.dat %>%
  pivot_longer(
    cols = c("Treatment Average", "Counterfactual Average", "Control Average", "att"),
    names_to = "category") %>%
  filter(category %in% c("Treatment Average", "Control Average", "Counterfactual Average"))

figure_limit_raw <- data.frame(out$time, out$Y.tr, out$Y.co)
figure_limit_raw <- cbind(figure_limit_raw,as.data.frame(day))

figure_limit_raw <- figure_limit_raw %>%
  pivot_longer(
    cols = c("AIR.DE":"WCH.DE"),
    names_to = "ticker")

figure_limit_raw$treatment <- ifelse(figure_limit_raw$ticker %in% t_seq, "Treated Raw Data", "Controls 
```

Create Figure.

```
ggplot()+
  geom_line(data=figure_limit_raw, aes(x=day, y=value, group=ticker,color= factor(treatment)), lwd=0.5, 
  geom_line(data=figure_limit, aes(x=day, y=value, group=category,color= factor(category)), lwd=1, key_g
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1,  color = "black")+
  ggtitle("")+
  xlab("")+
  ylab("CAR %")+
  labs(color=NULL)+
  scale_color_manual(name = "", values = c(`Treatment Average`="#3C93C2", `Counterfactual Average` = "#
                     breaks = c("Treatment Average","Counterfactual Average","Control Average", "Treate
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed",`Announcement Day` = "s
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 13),
    legend.position = "bottom"
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1),
         linetype= guide_legend(byrow = T, order = 2, ncol = 1))
```

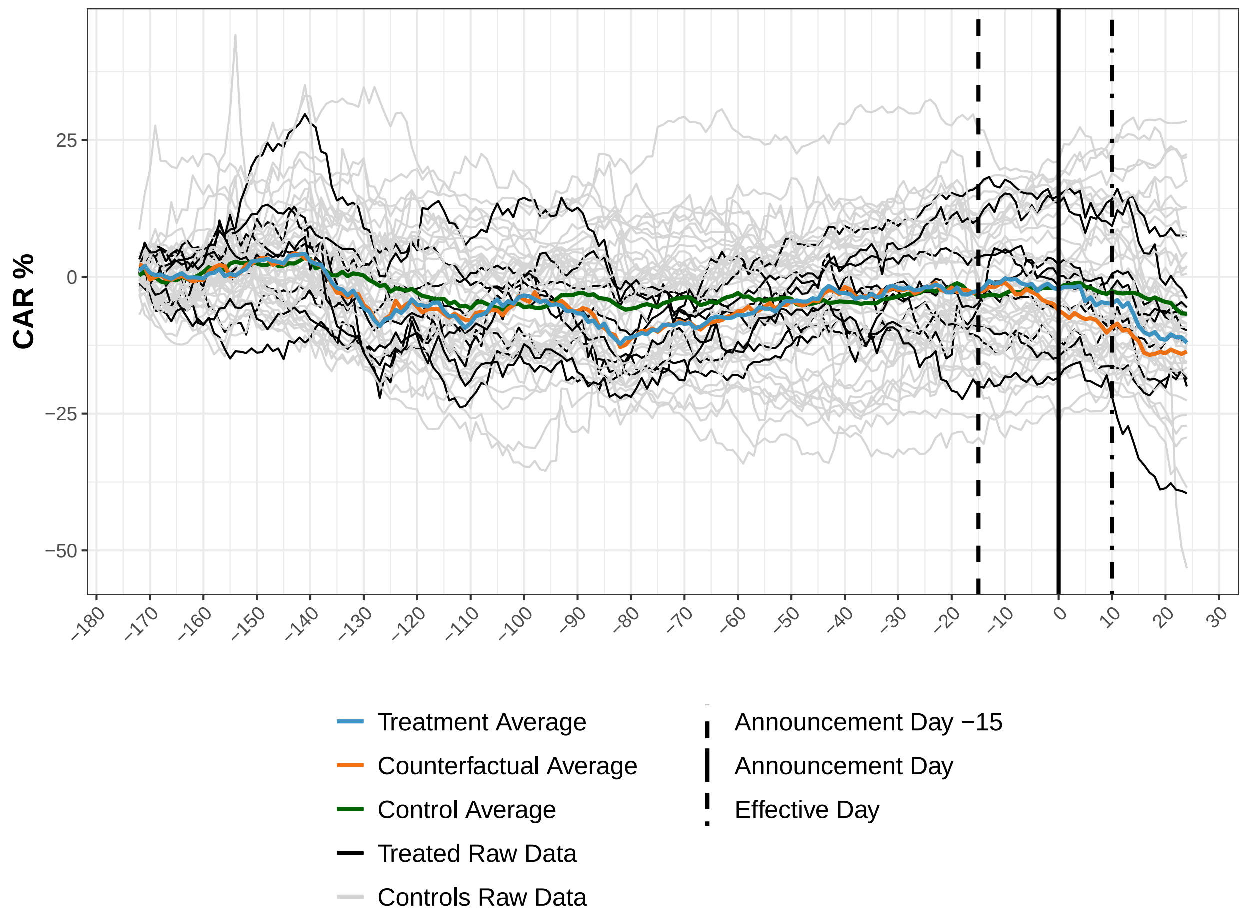


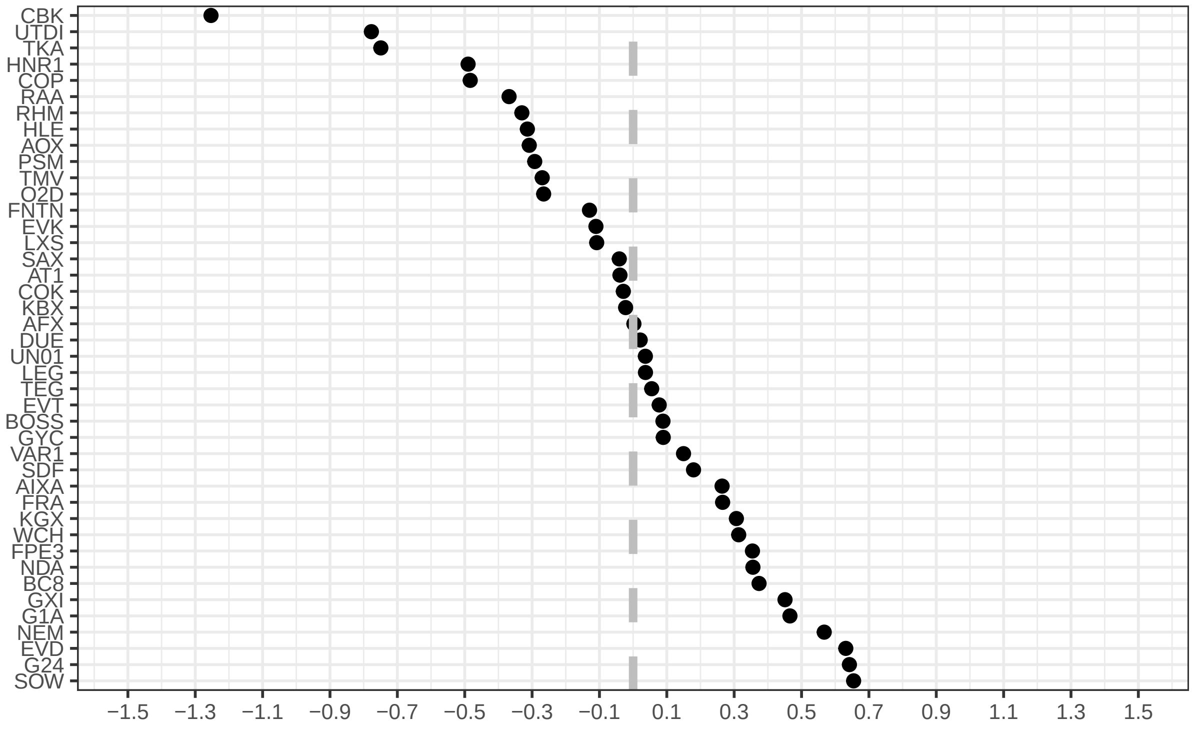
Figure 4: Unit weights

Data preparation.

```r
weights <- as.data.frame(out$wgt.implied[,8])
weights <- cbind(ticker = rownames(weights), weights)
rownames(weights) <- 1:nrow(weights)
weights <- weights %>%
  rename(weight = `out$wgt.implied[, 8]`) %>%
  mutate(Firm = gsub("\\..*", "", ticker) ) %>%
  select(weight, Firm)
```

Create figure.

```r
ggplot()+
  geom_point(data = weights, aes(x=weight, y=reorder(Firm, -weight)), size=2)+
  geom_vline(xintercept = 0,linetype="dashed", lwd=1.5, color = "grey")+
  xlab("Weight")+
  ylab("")+
  theme_bw()+
  theme(
    axis.title.x = element_text(color="black", size=15, face="bold", margin = margin(t = 30, r = 0, b =
    axis.text = element_text(size = 8),
    legend.position = "none"
  )+
  scale_x_continuous(breaks = seq(from = -1.5,to = 1.5, by = 0.2), limits = c(-1.5,1.5))
```



Weight

Figure 5: Average CAR of the treated and it's counterfactual

Data preparation.

```
figure_count_d <- out.dat %>%
  pivot_longer(
    cols = c("Treatment Average", "Counterfactual Average", "Control Average", "att"),
    names_to = "category") %>%
  filter(category %in% c("Treatment Average", "Counterfactual Average", "Control Average"))

figure_count_d$category <- factor(figure_count_d$category, levels = c("Control Average", "Treatment Ave
```

Create figure.

```
ggplot(data=figure_count_d)+
  geom_line(aes(x=day, y=value, color=category, group=category), lwd=1, key_glyph = "smooth")+
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1,  color = "black")+
  ggtitle("")+
  xlab("")+
  ylab("CAR %")+
  labs(color=NULL)+
  scale_color_manual(name = "", values = c(`Treatment Average` ="#3C93C2", `Counterfactual Average` = "#
                     breaks = c("Treatment Average","Counterfactual Average","Control Average"))+
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed",`Announcement Day` = "so
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom"
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1),
         linetype= guide_legend(byrow = T, order = 2, ncol = 1))
```

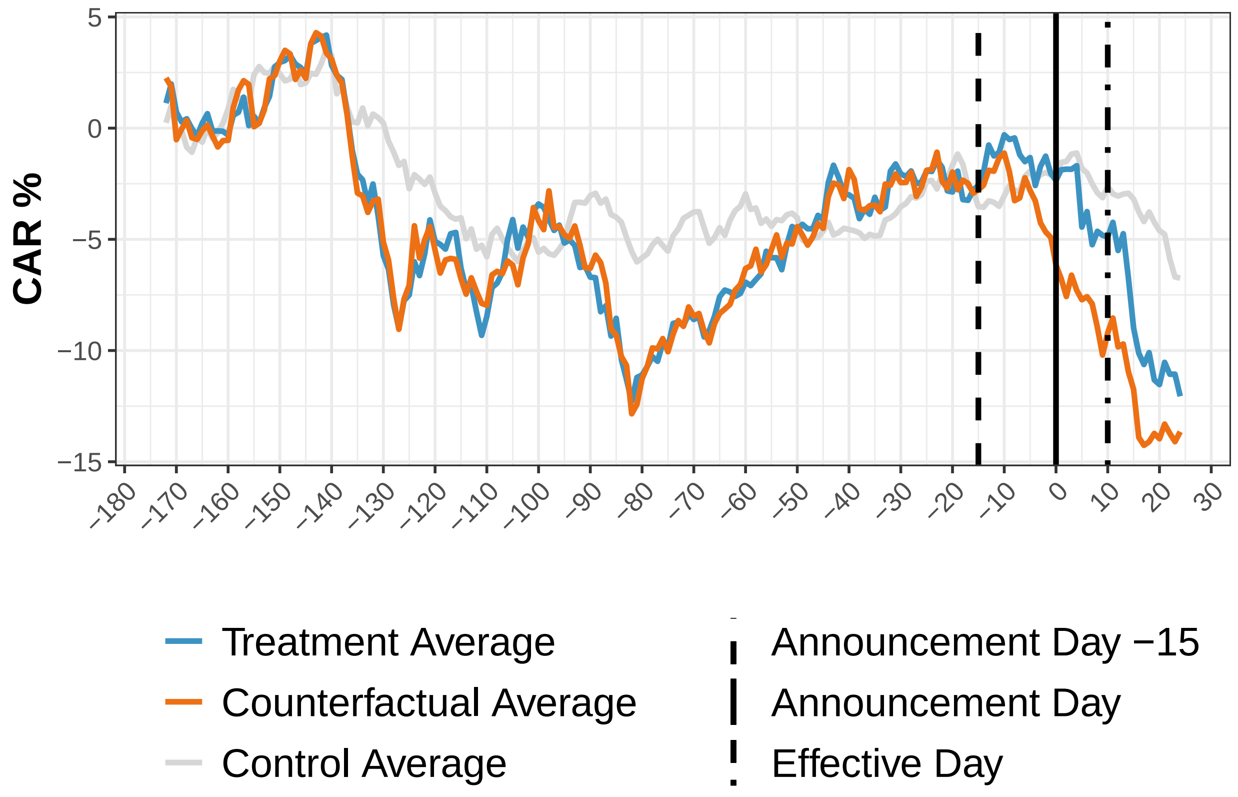


Figure 6: ATT and trading volume

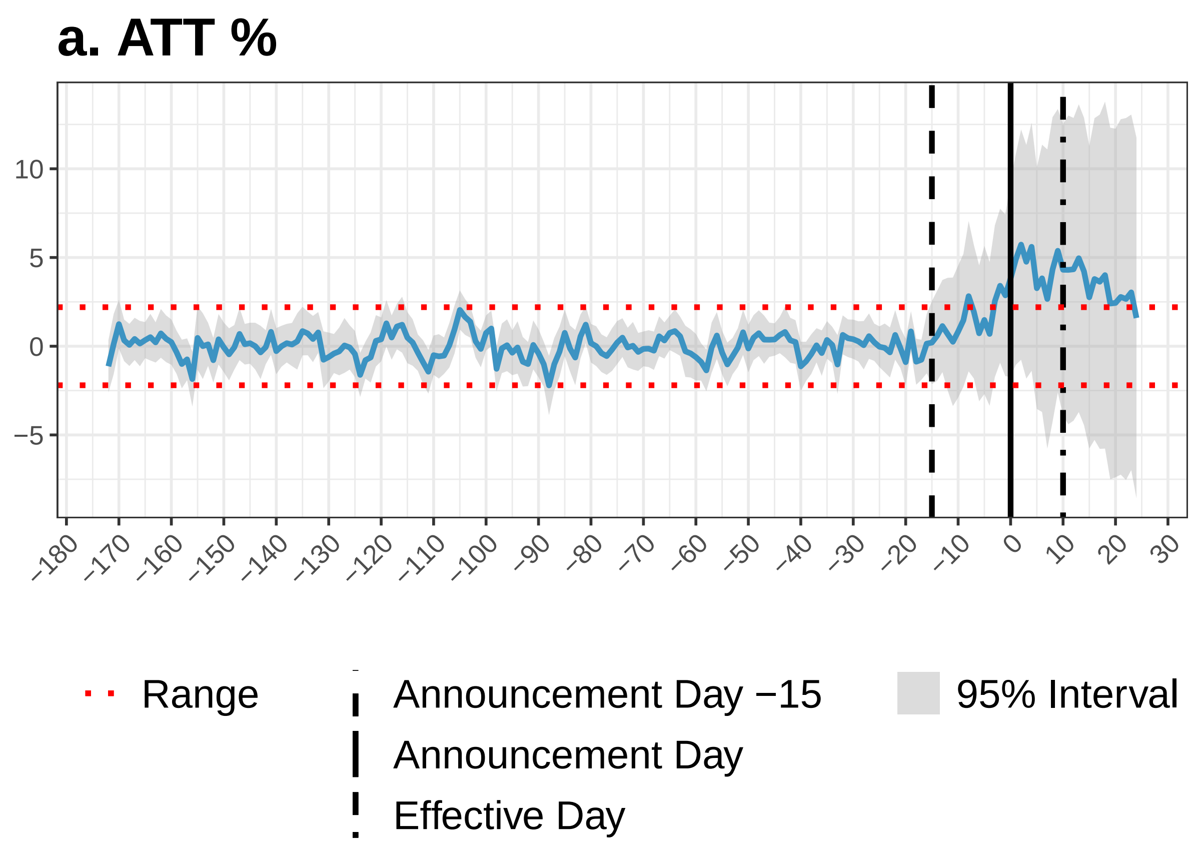
Create the two sub-figures of figure 6 and create a plot to extract the legend valid for both sub-figures.

Prepare data frame with est.att = inference for att, the att and time for subfigure 6a.

```r
figure_att_b <- data.frame(inf=out$est.att, att=out$att, time=out$time)
figure_att_b <- figure_att_b %>%
  rename(lower = inf.CI.lower,
         upper = inf.CI.upper) %>%
  select(lower, upper, att, time)
figure_att_b<- cbind(figure_att_b,as.data.frame(day))
```

Create sub-figure 6a: ATT%.

```r
figure_att <- ggplot()+
  geom_ribbon(data = figure_att_b, aes(x = day, ymin = lower, ymax = upper, fill = "95% Interval"), alp
  geom_line(data = figure_att_b, aes(x=day, y=att),color = "#3C93C2", lwd=1)+
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1,  color = "black")+
  geom_hline(aes(yintercept = 2.2, color = "Range"),lwd=1, linetype = "dotted")+
  geom_hline(aes(yintercept = -2.2, color = "Range"),lwd=1, linetype ="dotted")+
  ggtitle("a. ATT %")+
  xlab("")+
  ylab("")+
  labs(color=NULL)+
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed",`Announcement Day` = "s
                        breaks = c("Announcement Day -15","Announcement Day","Effective Day", "Range"))
  scale_color_manual(name = "", values = c(Range = "red"))+
  scale_fill_manual (name = "", values = c(`95% Interval` = "darkgrey"))+
  theme_bw()+
  theme(
    plot.title = element_text(size = 20, face = "bold"),
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom"
  )+
    guides(color = guide_legend(byrow = T, order = 1, ncol = 1),
           linetype= guide_legend(byrow = T, order = 2, ncol = 1))

figure_att
```



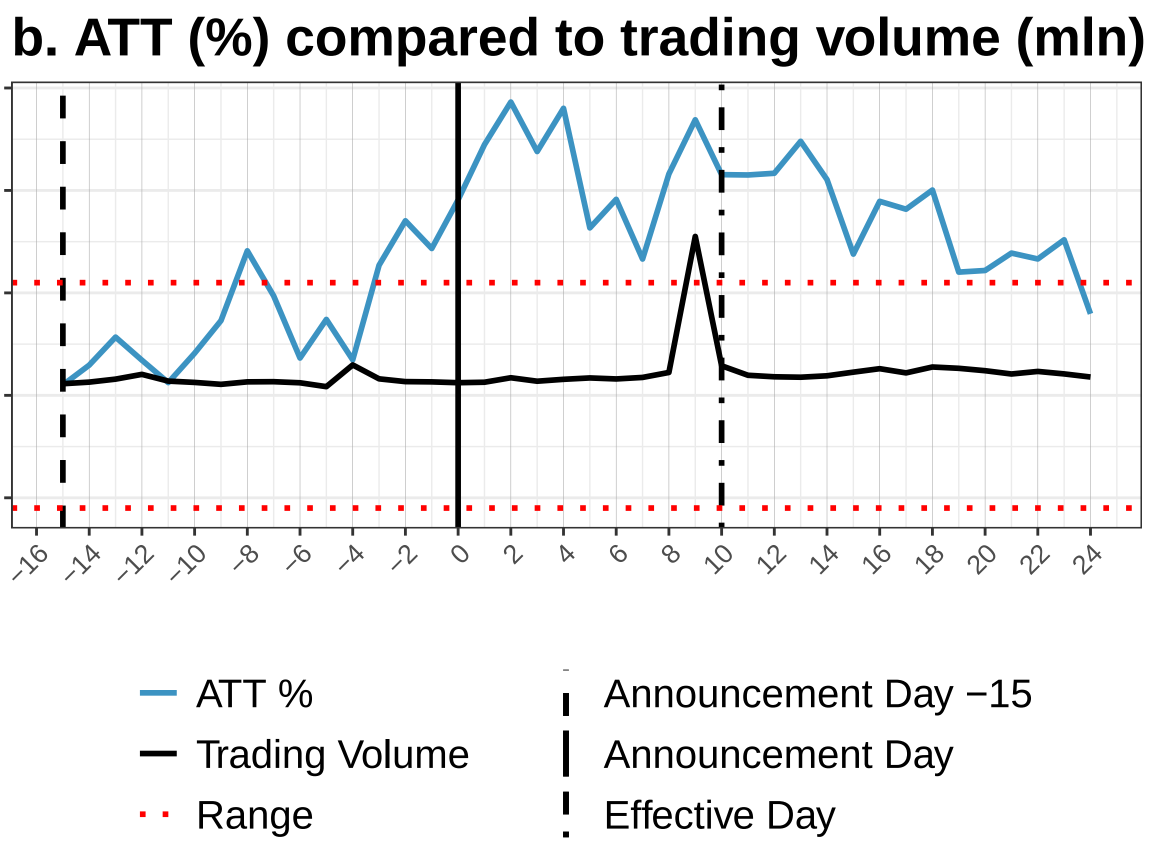
Data preparation for figure 6b: ATT% compared to trading volume (mln).

Note: "vol" is needed later for figure 8: ATT% compared to trading volume (mln) in the long run.

```r
volume <- Stock_data %>%
  select(ticker, date, volume, treatment) %>%
  mutate(date = as.Date(date), volume = (volume/1000000)) %>%
  filter(date >= as.Date("2021-01-01") & date <= as.Date("2021-10-08") & treatment == 1)

volume <- left_join(volume, figure_limit_raw, by = c("date" = "out.time", "ticker")) #ad days from data

vol <- volume %>%
  filter(treatment.x == 1) %>%
  select(day, volume, ticker) %>%
  group_by(day) %>%
  summarise(volume = mean(volume))

# Shorten the data to the treatment period
figure_att_b_short <- figure_att_b %>%
  filter(day >= -15)

vol_short <- vol %>%
  filter(day >= -15)
```

Create sub-figure 6b: ATT% compared to trading volume (mln).

```
figure_att_vol <- ggplot()+
  geom_line(data = figure_att_b_short, aes(x=day, y=att, color="ATT %", linetype="att"), lwd=1, show.leg
  geom_line(data = vol_short, aes(x=day, y=volume, color="Trading Volume", linetype="vol"), lwd=1, show
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1)+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1)+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1)+
  geom_hline(aes(yintercept = c(-2.2, 2.2), color = "Range"), linetype = "dotted", lwd=1)+
  ggtitle("b. ATT (%) compared to trading volume (mln)")+
  xlab("")+
  ylab("")+
  labs(color=NULL)+
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed", `Announcement Day` = "
                        breaks = c("Announcement Day -15","Announcement Day","Effective Day"))+
  scale_color_manual(name="", values=c(Range="red", `ATT %`="#3C93C2", `Trading Volume` ="black"), breal
  theme_bw()+
  theme(
    panel.grid.major.x = element_line( linewidth = 0.1, color="darkgrey"),
    plot.title = element_text(size = 20, face = "bold"),
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom"
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1, override.aes =
                                list(linetype=c("solid","solid","dotted"),
                                     lwd=c(1,1,1))),
         linetype = guide_legend(byrow = T, order = 2, ncol = 1))

figure_att_vol
```



Calculating the mean trading volume for the training period.

```
tab_vol <- vol %>%
  filter(day <= -15)
mean(tab_vol$volume)
```

```
## [1] 0.3982061
```

Create plot to capture the joint legend.

```
leg_full <- ggplot()+
  geom_ribbon(data = figure_att_b, aes(x = day, ymin = lower, ymax = upper, fill = "95% Interval"), alph
  geom_line(data = figure_att_b, aes(x=day, y=att, color="ATT %", linetype="att"), lwd=1, show.legend =
  geom_line(data = vol, aes(x=day, y=volume, color="Trading Volume", linetype="vol"), lwd=1, show.legen
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1)+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1)+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1)+
  geom_hline(aes(yintercept = c(-2.2, 2.2), color = "Range"), linetype = "dotted")+
  ggtitle("a.")+
  xlab("")+
  ylab("")+
  labs(color=NULL)+
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed", `Announcement Day` = "
                        breaks = c("Announcement Day -15","Announcement Day","Effective Day"))+
  scale_color_manual(name="", values=c(Range="red", `ATT %`="#3C93C2", `Trading Volume` ="black"), breal
  scale_fill_manual(name = "", values = c(`95% Interval` = "darkgrey"))+
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom",
    plot.margin=grid::unit(c(0.5,0.5,0.5,-15), "mm")
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1, override.aes =
                              list(linetype=c("solid","solid","dotted"),
                                   lwd=c(1,1,1))),
         linetype = guide_legend(byrow = T, order = 2, ncol = 1))
```

Capture the legend from combined plot and create the entire plot.

```
leg <- get_legend(leg_full)

rm_legend <- function(p){p + theme(legend.position = "none")}
plots <- ggarrange(rm_legend(figure_att), rm_legend(figure_att_vol), nrow = 2)
figure_att_volume <- ggarrange(plots, legend.grob = leg, legend="bottom")

figure_att_volume
```

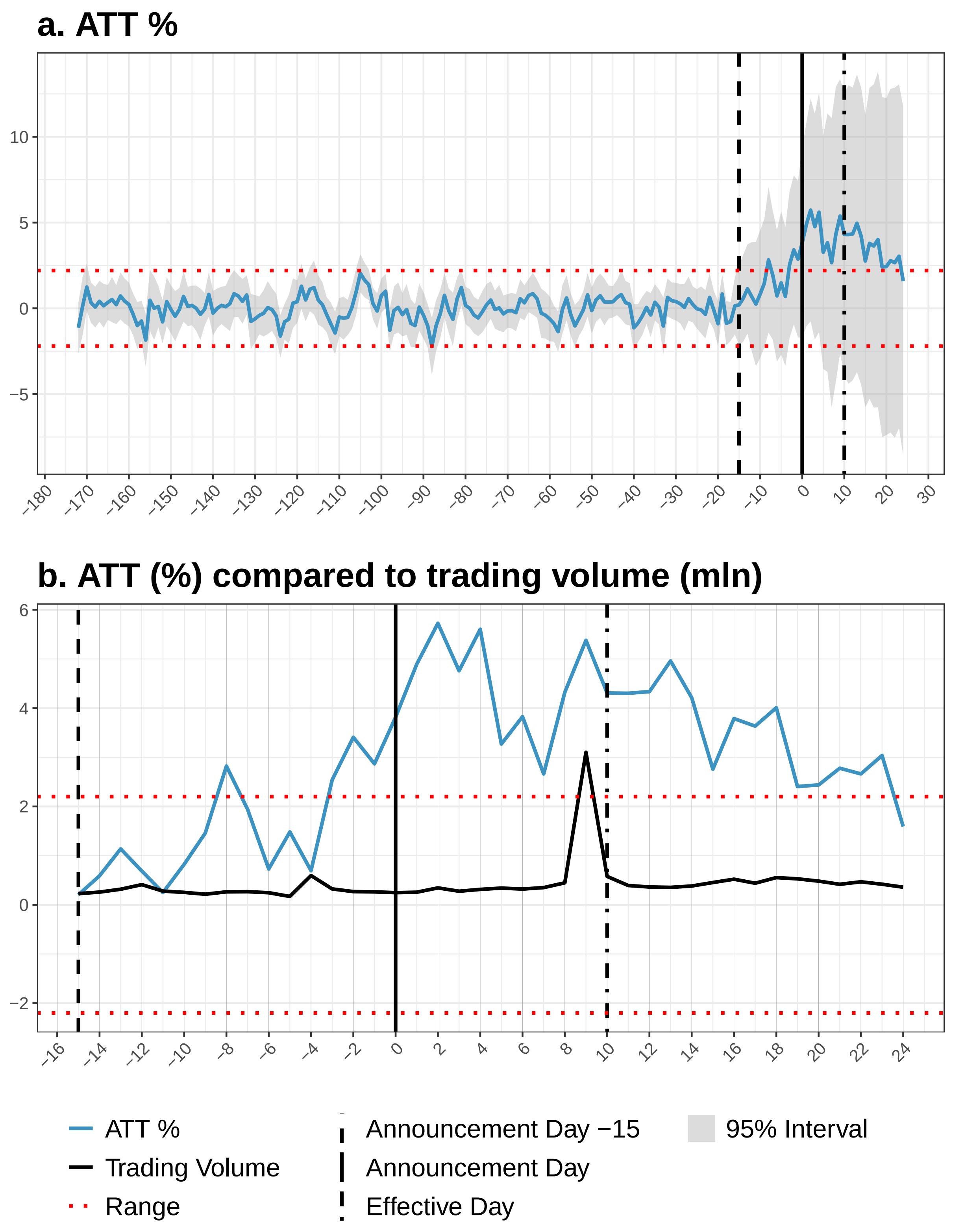


Figure 7: Selection of the CAPM-period

Load data from Yahoo! Finance (https://finance.yahoo.com/) and prepare it.

```r
ma_index <- c("^GDAXIP","^MDAXIP")
figure_app_data <- lapply(ma_index, function(sym) {
  getSymbols(sym, from="2018-01-01", to="2021-12-31", auto.assign=FALSE)
})
figure_app_data <- as.data.frame(figure_app_data)
figure_app_data <- figure_app_data %>%
  select(GDAXIP.Close, MDAXIP.Close) %>%
  rename(`DAX Price Index` = GDAXIP.Close, `MDAX Price Index` = MDAXIP.Close)
figure_app_data <- cbind(date = rownames(figure_app_data), figure_app_data)
rownames(figure_app_data) <- 1:nrow(figure_app_data)
figure_app_data$date <- as.Date(figure_app_data$date)

figure_app_data <- figure_app_data %>%
  pivot_longer(
    cols = c("DAX Price Index", "MDAX Price Index"),
    names_to = "index")
```

Create figure.

```r
ggplot(data=figure_app_data)+
  geom_rect(aes(xmin = as.Date("2020-07-01"), xmax = as.Date("2021-08-13"), ymin = 0, ymax = 20000, fil
  geom_vline(aes(xintercept = as.Date("2020-02-20"), linetype="Start Corona Crash"),lwd=1, color = "bla
  geom_line(aes(x=date, y=value, color=index, group=index), lwd=1, key_glyph = "smooth")+
  ggtitle("")+
  xlab("")+
  ylab("Closing Price (Euro)")+
  scale_color_manual(name = "", values = c(`DAX Price Index` ="#3C93C2", `MDAX Price Index` = "#ED7014"
                     breaks = c("DAX Price Index","MDAX Price Index"))+
  scale_linetype_manual(name = "", values = c(`Start Corona Crash` ="dashed"),
                        breaks = c("Start Corona Crash"))+
  scale_fill_manual(name = "", values = c(`CAPM Period` ="grey84"),
                     breaks = c("CAPM Period"))+
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=15, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 14),
    legend.position = "bottom"
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1),
         linetype= guide_legend(byrow = T, order = 2, ncol = 1))
```

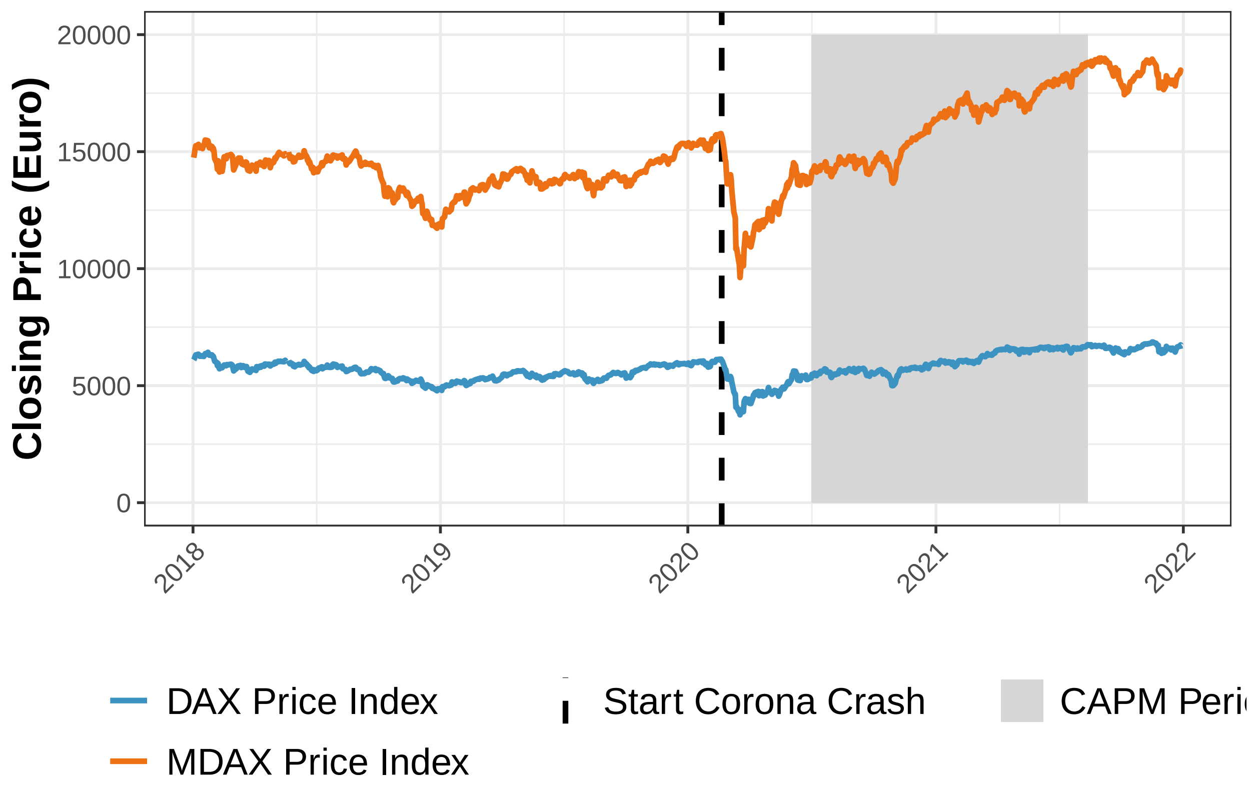
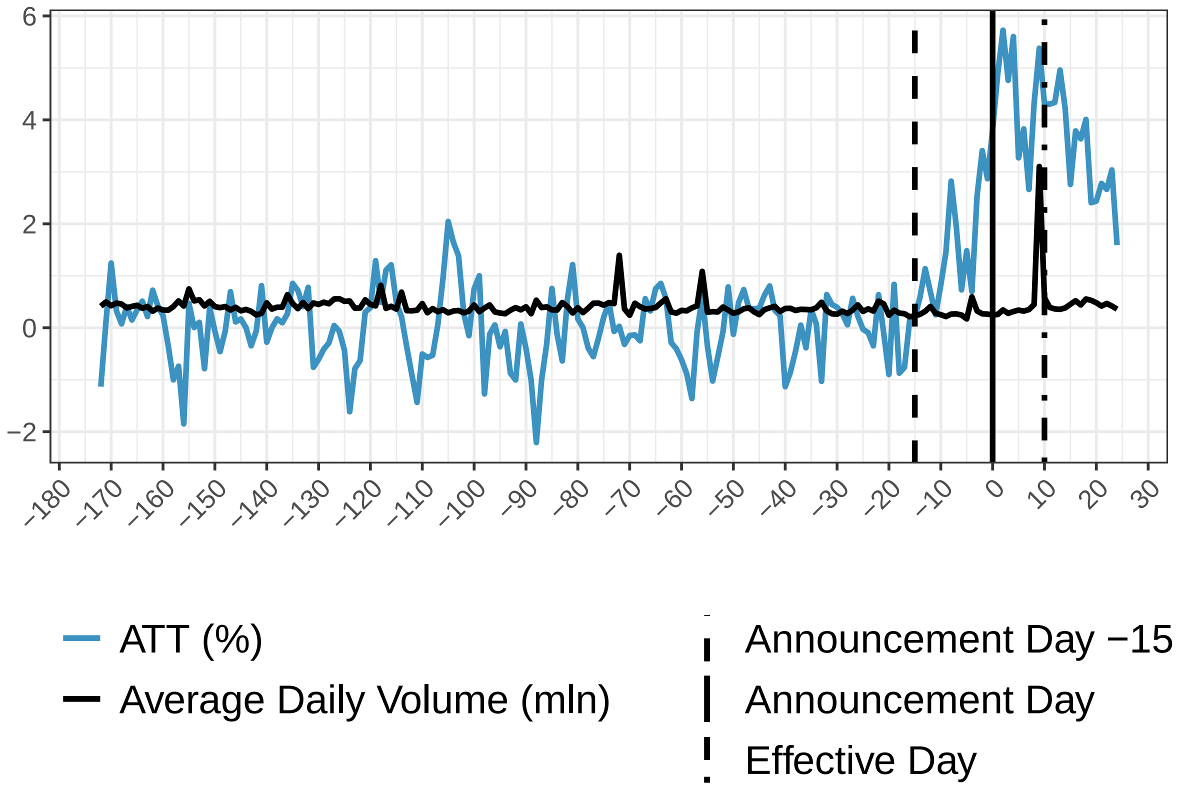


Figure 8: ATT (%) compared to trading volume (mln) in the long run

```
ggplot()+
  geom_line(data = figure_att_b, aes(x=day, y=att, color= "ATT (%)"), lwd=1)+
  geom_line(data = vol, aes(x=day, y=volume, color="Average Daily Volume (mln)"), lwd=1, key_glyph = "sm
  scale_x_continuous(n.breaks = 20)+
  geom_vline(aes(xintercept = -15, linetype="Announcement Day -15"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 0, linetype="Announcement Day"),lwd=1, color = "black")+
  geom_vline(aes(xintercept = 10, linetype="Effective Day"),lwd=1,  color = "black")+
  ggtitle("")+
  xlab("")+
  ylab("")+
  labs(color=NULL)+
  scale_linetype_manual(name = "", values = c(`Announcement Day -15` = "dashed",`Announcement Day` = "so
                      breaks = c("Announcement Day -15","Announcement Day","Effective Day", "Range"))
  scale_color_manual(name = "", values = c(`ATT (%)` = "#3C93C2", `Average Daily Volume (mln)` = "black
  theme_bw()+
  theme(
    axis.title.y = element_text(color="black", size=20, face="bold"),
    axis.text = element_text(size = 10),
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.text = element_text(size = 15),
    legend.position = "bottom",
  )+
  guides(color = guide_legend(byrow = T, order = 1, ncol = 1),
         linetype= guide_legend(byrow = T, order = 2, ncol = 1))
```



Descriptive Analysis

Data Preparation

Create a vector with file names of files stored in the working directory and load them.

Make sure, that only the files from the folder “Data_1” are stored in this working directory.

Please note, for ECV, HOT, MOR, NDX1 and SAE are no data available, as we excluded them as described in section 3.4.

```
setwd("C:/Users/jfur2/OneDrive/Dokumente/Masterarbeit/GSC/Data_1")

list_csv_files <- list.files()
desc_stat <- readr::read_csv(list_csv_files, id = "file_name")
```

```
## Rows: 14242 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (11): time, open, high, low, close, P/B ratio, Basic EPS, EBITDA margin ...
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Do some basic data preparation.

```
# Convert UNIX-time to date and remove time stamp
desc_stat$time <- as_datetime(desc_stat$time)
desc_stat$time <- format(as.POSIXct(desc_stat$time,
                              format = '%Y-%m-%d %H:%M:%S'),
                  format = '%Y-%m-%d')

# Exclude data smaller than 2021-01-01 and bigger than 2021-09-30
desc_stat <- desc_stat[!(desc_stat$time <= "2021-01-01"),]
desc_stat <- desc_stat[!(desc_stat$time >= "2021-09-30"),]

# Change name of company to the company ticker
desc_stat$file_name <- sub(".*? ", "", desc_stat$file_name)
desc_stat$file_name <- gsub(",.*", "", desc_stat$file_name)
```

Load and add file with sector and treatment information of the companies (folder "Data_2"), delete irrelevant columns and rename relevant columns.

```
sector <- read_excel("C:/Users/jfur2/OneDrive/Dokumente/Masterarbeit/GSC/Data_2/v2_Branche and Industry
desc_stat <- left_join(desc_stat, sector, by= c("file_name"="Firmtag"))
desc_stat <- desc_stat %>%
  select(-(3:6), -(10), -(15))

cnames <- c("ticker", "date", "pbratio", "basiceps", "ebitdamargin", "mktcap", "divyield", "name", "sec
colnames(desc_stat) <- cnames
```

Enlarge data manually with data from individual quarterly reports and macrotrends (https://www.macrotrends.net).

```
## WCH: https://www.wacker.com/cms/en-us/about-wacker/investor-relations/financial-reports/financial-re
# Q1 -21
desc_stat$basiceps[1998] <- 2.15
desc_stat$ebitdamargin[1998] <- 18.1

# Q2 -21
desc_stat$basiceps[2011] <- 3.56
desc_stat$ebitdamargin[2011] <- 21.8

# Q3 -21
desc_stat$basiceps[2024] <- 4.9
desc_stat$ebitdamargin[2024] <- 26.1

## G24: https://www.scout24.com/en/investor-relations/financial-reports-presentations
# Q1 -21
desc_stat$basiceps[828] <- 0.25
desc_stat$ebitdamargin[828] <- 55.7649

# Q2 -21
desc_stat$basiceps[841] <- 0.25
```

```r
desc_stat$ebitdamargin[841] <- 51.094

# Q3 -21
desc_stat$basiceps[854] <- 0.29
desc_stat$ebitdamargin[854] <- 53.45589

## O2D: https://www.telefonica.de/investor-relations/publikationen/finanzpublikationen.html
# Q1 -21
desc_stat$basiceps[1335] <- -0.01
desc_stat$ebitdamargin[1335] <- 29.6

# Q2 -21
desc_stat$basiceps[1348] <- -0.03
desc_stat$ebitdamargin[1348] <- 32.2

# Q3 -21
desc_stat$basiceps[1361] <- 0.05
desc_stat$ebitdamargin[1361] <- 44.6

## AFX: https://www.zeiss.com/meditec-ag/investor-relations/reports-publications.html
#Q1 - 21
desc_stat$basiceps[13] <- 0.6

# Q2 -21
desc_stat$basiceps[26] <- 0.92
desc_stat$ebitdamargin[26] <- 26.63876 #proxy (mean EBITDA margin Q1 and Q3 2021)

# Q3 -21
desc_stat$basiceps[39] <- 0.6

## EVT: https://www.macrotrends.net/stocks/charts/EVO/evotec-ag/ebitda-margin
# Q1 - 21
desc_stat$ebitdamargin[633] <- 7.5

# Q1 - 21
desc_stat$ebitdamargin[646] <- 6.83

# Q1 - 21
desc_stat$ebitdamargin[659] <- 6.75

## HNR1: https://www.macrotrends.net/stocks/charts/HVRRY/hannover-ruck-se/ebitda-margin
# Q1 - 21
desc_stat$ebitdamargin[1023] <- 6.38

# Q1 - 21
desc_stat$ebitdamargin[1036] <- 7.79

# Q1 - 21
desc_stat$ebitdamargin[1049] <- 7.07
```

Exclude irrelevant firms.

Further, scale market capitalization and calculate the log of the P/B ratio.

```r
firmex <- c("PAH3", "BEI", "AG1", "LHA")
desc_stat <- desc_stat %>%
  filter(!(ticker %in% firmex))

desc_stat <- desc_stat %>%
  select(ticker, date, pbratio, basiceps, ebitdamargin, mktcap, divyield, sector, treatment) %>%
  mutate(mktcap = mktcap/1000000000, pbratio = log(pbratio))
```

Create Descriptive Statistics

Setting variables of interest for descriptive statistic.

```r
mycontrols <- tableby.control(test = F,
                              numeric.stats=c("mean", "medianrange", "q1q3", "sd", "N" ),
                              digits = 2)
```

To get a more precise analysis we need to exclude firms with missing data.

```r
tab1 <- summary(tableby(treatment ~ mktcap + pbratio, data = desc_stat, control=mycontrols), text = NULL
tab1 <- as.data.frame(tab1)

tab2 <- summary(tableby(treatment ~ ebitdamargin, subset= !(ticker %in% c("CBK", "PUM", "RAA", "SY1")),
tab2 <- as.data.frame(tab2)

tab3 <- summary(tableby(treatment ~ divyield, subset= !(ticker %in% c("AFX", "G24", "O2D", "SAX",  "SY1
tab3 <- as.data.frame(tab3)

tab4 <- summary(tableby(treatment ~ basiceps, subset= !(ticker %in% c("SY1")), data = desc_stat, contro
tab4 <- as.data.frame(tab4)
```

Create new data sets with CAR and create statistics for different time periods.

```r
## Day -15 to 0
# Get relevant data from "Stock_data"
CAR_desc_p1 <- Stock_data%>%
  filter(date >= (as.Date("2021-08-16")) & date <= (as.Date("2021-09-04")))

# Calculate cumulative abnormal return
CAR_desc_p1 <- CAR_desc_p1 %>%
  group_by(ticker) %>%
  mutate(CAR_p1 = cumsum((AR*100)))

# Create descriptive statistic
tab5 <- summary(tableby(treatment ~ CAR_p1, data = CAR_desc_p1, control=mycontrols), text = NULL)
tab5 <- as.data.frame(tab5)

## Day 0 to 10
# Get relevant data from "Stock_data"
CAR_desc_p2 <- Stock_data %>%
  filter(date >= (as.Date("2021-09-04")) & date < (as.Date("2021-09-20")))
```

```r
# Calculate cumulative abnormal return
CAR_desc_p2 <- CAR_desc_p2 %>%
  group_by(ticker) %>%
  mutate(CAR_p2 = cumsum((AR*100)))

# Create descriptive statistic
tab6 <- summary(tableby(treatment ~ CAR_p2, data = CAR_desc_p2, control=mycontrols), text = NULL)
tab6 <- as.data.frame(tab6)

# Day 10 to 24
# Get relevant data from "Stock_data"
CAR_desc_p3 <- Stock_data %>%
  filter(date >= (as.Date("2021-09-20")) & date <= (as.Date("2021-10-08")))

# Calculate cumulative abnormal return
CAR_desc_p3 <- CAR_desc_p3 %>%
  group_by(ticker) %>%
  mutate(CAR_p3 = cumsum((AR*100)))

# Create descriptive statistic
tab7 <- summary(tableby(treatment ~ CAR_p3, data = CAR_desc_p3, control=mycontrols), text = NULL)
tab7 <- as.data.frame(tab7)
```

Prepare these newly created data sets.

```r
# Rename columns of created data frames
dfs <- c("tab1", "tab2", "tab3", "tab4", "tab5", "tab6", "tab7")
for(df in c("tab1", "tab2", "tab3", "tab4", "tab5", "tab6", "tab7")) {
  data.table::setnames(get(df),  c("Measure", "Control Group", "Treatment Group", "Total"))
}

# Merge data frames
tab_fund <- rbind(tab1, tab2, tab3, tab4, tab5, tab6, tab7)

# Create new a vector with key variables names and add it to the data set "tab_fund"
Key_Variable <- c(rep("Market Cap. (bn €)", 6),rep("P/B Ratio (log %)", 6), rep("EBITDA Margin (%)", 6)
                  rep("CAR (%) [-15,0]", 6), rep("CAR (%) [0,10]", 6), rep("CAR (%) [10,24]", 6))

tab_fund <- cbind(tab_fund, Key_Variable)

tab_fund <-  tab_fund %>%
  filter(!(`Control Group`==""))

# Restructure data for a better overview
table <- gather(tab_fund, Group, measurement,`Control Group`: Total, factor_key=TRUE)
table2 <- table %>%
  mutate(Measure = factor(Measure, levels = unique(Measure))) %>%
  mutate(Key_Variable = factor(Key_Variable, levels = unique(Key_Variable))) %>%
  spread(Measure, measurement) %>%
  mutate(id = 1:24) %>%
  mutate (Group = paste(id, Group, sep = ". ")) %>%
  select(-("id"))
```

Define table parameters and store the final table.

```
tbl <- create_table(table2, first_row_blank = TRUE, header_bold = T, borders = c("top", "bottom")) %>%
  stub(c("Key_Variable", "Group")) %>%
  define(Key_Variable, blank_after = T, label_row = TRUE) %>%
  define(Group, indent = .25) %>%
  define(Mean, label = "Mean", align = "center") %>%
  define(`Median (Range)`, label = "Median (Range)", align = "center") %>%
  define(`Q1, Q3`, label = "Q1, Q3", align = "center") %>%
  define(SD, label = "SD", align = "center") %>%
  define(N, label = "N", align = "center")

# Create the report
rpt <- create_report(output_type = "RTF",
                     font = "times", font_size = 11, file_path = "C:/Users/jfur2/OneDrive/Dokumente/Mas
  add_content(tbl)

# Write report
write_report(rpt)
```

```
## # A report specification: 1 pages
## - file_path: 'C:/Users/jfur2/OneDrive/Dokumente/Masterarbeit/GSC/Desc_stat/desc_analysis.rtf'
## - output_type: RTF
## - units: inches
## - orientation: landscape
## - margins: top 0.5 bottom 0.5 left 1 right 1
## - line size/count: 9/42
## - content:
## # A table specification:
## - data: data.frame 'table2' 24 rows 7 cols
## - show_cols: all
## - use_attributes: all
## - stub: Key_Variable Group align='left'
## - define: Key_Variable dedupe='TRUE'
## - define: Group
## - define: Mean 'Mean' align='center'
## - define: Median (Range) 'Median (Range)' align='center'
## - define: Q1, Q3 'Q1, Q3' align='center'
## - define: SD 'SD' align='center'
## - define: N 'N' align='center'
```

Sector Analysis

Prepare Data.

```
sector <- desc_stat %>%
  select(ticker, sector, treatment) %>%
  distinct(ticker, sector, treatment)
```

Create sector analysis.

```
tab8 <- summary(tableby(treatment ~ sector, data = sector, control=mycontrols), text = NULL)
tab8 <- as.data.frame(tab8)
tab8 <- tab8 %>%
  rename(sector=1) %>%
  filter(!(`0 (N=42)`=="")) %>%
  arrange(desc(`1 (N=9)`)) %>%
  mutate(id = 1:9) %>%
  mutate (sector = paste(id, sector, sep = ". ")) %>%
  select(-("id"))
```

Define table parameters and store the table.

```
tbl_sector <- create_table(tab8, first_row_blank = T, header_bold = T, borders = c("top", "bottom")) %>%
  define(sector, label= "Sector", blank_after = F, label_row = F) %>%
  define(`0 (N=42)`, label = "Control Group (N=42)", align = "center") %>%
  define(`1 (N=9)`, label = "Treatment Group (N=9)", align = "center") %>%
  define(`Total (N=51)`, label = "Total  (N=51)", align = "center")

# Create the report
rpt_sector <- create_report(output_type = "RTF",
                     font = "times", font_size = 12,  file_path = "C:/Users/jfur2/OneDrive/Dokumente/Ma
  add_content(tbl_sector)

# Write out report
write_report(rpt_sector)
```